RECEIVED
CENTRAL FAX CENTER
APR 24 2006

PATENT

MS146910.01/MSFTP119US

CERTIFICATE OF FACSIMILE TRANSMISSION

I hereby certify that this correspondence (along with any paper referred to as being attached or enclosed) is being faxed to 571-273-8300 on the date shown below to Mail Stop Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Date: 4-24-06

Christina M. Palamonsky

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent application of:

Applicants(s): Srivatsan Parthasarathy, *et al.*

Serial No: 09/604,987

Filing Date: June 28, 2000

Examiner: Tuan A. Vu

Art Unit: 2193

Title: BINDING BY HASH

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Dear Sir:

Appellants submit this brief in connection with an appeal of the above-identified patent application. A credit card payment form is filed concurrently herewith in connection with all fees due regarding this appeal brief. In the event any additional fees may be due and/or are not covered by the credit card, the Commissioner is authorized to charge such fees to Deposit Account No. 50-1063 [MSFTP119US].

04/25/2006 MBINAS 00000024 09604987
01 FC:1402 500.00 OP

I. Real Party in Interest (37 C.F.R. §41.37(c)(1)(i))

The real party in interest in the present appeal is Microsoft Corporation, the assignee of the present application.

II. Related Appeals and Interferences (37 C.F.R. §41.37(c)(1)(ii))

Appellants, appellants' legal representative, and/or the assignee of the present application are not aware of any appeals or interferences which may be related to, will directly affect, or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. Status of Claims (37 C.F.R. §41.37(c)(1)(iii))

Claims 1-8, 10-18, 20-27 and 29-35 stand rejected by the Examiner. The rejection of claims 1-8, 10-18, 20-27 and 29-35 is being appealed.

IV. Status of Amendments (37 C.F.R. §41.37(c)(1)(iv))

No amendments were made after the Final Office Action dated October 7, 2005.

V. Summary of Claimed Subject Matter (37 C.F.R. §41.37(c)(1)(v))

A. Independent Claim 1

Independent claim 1 recites a method for facilitating integrity of an assembly employable by application programs during runtime, comprising: providing an assembly with an assembly manifest that contains a list of modules that make up the assembly; providing the assembly manifest with a hash of the contents of at least one module of the list of modules; providing the assembly manifest with a hash of a manifest of at least one other assembly that the assembly depends on; and comparing the hash retained in the assembly manifest with a hash of the at least one module obtained at runtime to identify whether a runtime version of the at least one module is substantially similar to a version utilized at build time of the assembly. (*See e.g.*, page 6, line 1-page 7, line 16, page 14, lines 8-23)

B. Independent Claim 10

Independent claim 10 recites A method for facilitating integrity of assemblies employable by application programs during runtime, comprising: providing an assembly with an assembly

manifest that contains a list of referenced assemblies that the assembly depends on; providing the assembly manifest with a hash of a manifest of at least one referenced assembly of the list of referenced assemblies; and analyzing the hash provided to the assembly manifest and a second hash of the manifest of the at least one referenced assembly computed at runtime to determine whether changes have been made to the at least one referenced assembly between runtime and at build time of the assembly. (*See e.g.*, page 6, line 1-page 7, line 16, page 14, lines 8-23)

C. **Independent Claim 18**

Independent claim 18 recites a computer readable medium having at least one computer executable component employable by an application program at runtime comprising: an assembly including an assembly manifest that contains a list of at least one referenced assembly that the assembly references, a first hash of a manifest of the at least one referenced assembly, a list of modules that make up the assembly and a hash of the contents of at least one module of the list of modules, the hash of the contents of the at least one module is utilized to control which versions of the modules are employed in connection with the assembly at runtime. (*See e.g.*, page 6, line 1-page 7, line 16, page 12, lines 19-22, page 14, lines 8-23)

D. **Independent Claim 22**

Independent claim 22 recites a computer readable medium having at least one computer executable component employable by an application program at runtime comprising: an assembly including an assembly manifest that contains a list of at least one referenced assembly that the assembly references and a hash of the contents of a manifest of the at least one referenced assembly, the hash is compared to a second hash produced at runtime to evaluate whether the at least one referenced assembly is a same version as the at least one referenced assembly utilized at build time of the assembly. (*See e.g.*, page 6, line 1-page 7, line 16, page 14, lines 8-23)

E. **Independent Claim 23**

Independent claim 23 recites a system for facilitating integrity of assemblies employable by application programs at runtime, the system comprising: a first component that provides an assembly manifest for an assembly, the assembly manifest having a list of modules making up

the assembly and a list of at least one referenced assembly that the assembly references; and a second component that provides the assembly manifest with a hash of at least one module of the list of modules and a hash of a manifest of the at least one referenced assembly, the hash of the at least one module is compared with a hash of the at least one module generated at runtime to identify changes in the content of the at least one module. (*See e.g.*, page 6, line 1-page 7, line 16, page 14, lines 8-23)

F. **Independent Claim 27**

Independent claim 27 recites a system for facilitating integrity of assemblies employable by application programs at runtime, the system comprising: a first component that provides an assembly manifest for an assembly, the assembly manifest having at least one referenced assembly, the at least one referenced assembly comprising a manifest; a second component that provides the assembly manifest with a hash of the manifest of the at least one referenced assembly; and a third component that compares the hash of the at least one referenced assembly in the assembly manifest with an actual hash value of the at least one referenced assembly to identify version changes. (*See e.g.*, page 6, line 1-page 7, line 16, page 14, lines 8-23)

G. **Independent Claim 30**

Independent claim 30 recites a system for facilitating integrity of an assembly employable by application programs at runtime, the system comprising: means for relating an assembly manifest having a list of at least one related assembly to an assembly (*See e.g.*, page 6, lines 1- 10), the at least one related assembly comprising a manifest; means for providing the assembly manifest with a hash of the manifest of the at least one related assembly (*See e.g.*, page 7, lines 5-15); and means for evaluating integrity of the assembly by comparing the hash value with a second hash value computed at runtime (*See e.g.*, page 6, lines 26- page 7, line 4).

The means for limitations described above are identified as limitations subject to the provisions of 35 U.S.C. §112 ¶6. The structures corresponding to these limitations are identified with reference to the specification and drawings in the above-noted parentheticals.

VI. Grounds of Rejection to be Reviewed (37 C.F.R. §41.37(c)(1)(vi))

A. Whether claims 1-8, 10-18, 20-27 and 29-35 are unpatentable under 35 U.S.C. §103(a) over Renaud *et al.* (U.S. 5,958,051), in view of Shaw (U.S. App. 2002/0026634), and further in view of Graunke *et al.* (U.S. 5,991,399) and Evans *et al.* (U.S. 5,805,899).

VII. Argument (37 C.F.R. §41.37(c)(1)(vii))

A. Rejection of Claims 1-8, 10-18, 20-27 and 29-35 Under 35 U.S.C. §103(a)

Claims 1-8, 10-18, 20-27 and 29-35 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Renaud *et al.* (U.S. 5,958,051), in view of Shaw (U.S. App. 2002/0026634), and further in view of Graunke *et al.* (U.S. 5,991,399) and Evans *et al.* (U.S. 5,805,899). It is respectfully requested that this rejection be withdrawn for at least the following reasons. The cited references, alone or in combination, fail to teach or suggest all limitations recited the subject claims.

> To reject claims in an application under §103, an examiner must establish a *prima facie* case of obviousness. A *prima facie* case of obviousness is established by a showing of three basic criteria. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, ***to modify the reference or to combine reference teachings***. Second, there must be a reasonable expectation of success. Finally, ***the prior art reference (or references when combined) must teach or suggest all the claim limitations.*** *See* MPEP §706.02(j). The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. *See In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991) (emphasis added).

The subject invention relates to validating that correct modules and assemblies are used for an application in a dynamically linked runtime environment. An assembly can consist of modules and other assemblies. A manifest can contain a list of modules and/or assemblies. A hash of modules and manifests are used to verify correct versions and identify changes to

modules and assemblies. An assembly manifest that contains a hash of a manifest of a referenced assembly can eliminate or reduce the need to validate the contents of the modules in the referenced assembly. In particular, independent claim 1 (and similarly independent claims 10, 18, 22, 23, 27 and 30) recites ***providing an assembly manifest with a hash of a manifest of at least one other assembly that the assembly depends on.***

Contrary to assertions in the Office Action dated October 7, 2005, the combination of Renaud *et al.*, Shaw, Graunke *et al.* and Evans *et al.* does not teach or suggest the aforementioned claimed features of appellants' invention. Renaud *et al* discloses a system for verifying the authenticity of data files by creating a signature file that contains a list of hashed identifiers for each file and a hashed signature of the signature file. The Examiner concedes that Renaud *et al.* does not "disclose a manifest with a hash of a manifest of one referenced assembly of the list of referenced assemblies." (*See* Office Action dated April 21, 2005, pg. 8). However, the Office Action incorrectly asserts that the site certificates of Renaud teach referenced assemblies. The site certificates merely provide a security layer between communicating servers. A site certificate identifies a server and servers employ these certificates to verify the authenticity of each server with which it is communicating. They do not specify referenced assemblies. Renaud *et al.* teaches a manifest containing a list of modules. Thus, Renaud *et al.* does not teach or suggest that the manifest is provided with a hash of a manifest of the at least one referenced assembly as recited in the subject claims.

Shaw fails to make up for the aforementioned deficiencies of Renaud *et al.* Shaw relates to a system for secure downloading of data. (*See* Abstract). Shaw discloses a table of hashes of segments of application code. (*See* Para 0034). Accordingly, Shaw discloses a hash of a code segment or module, and does not teach a hash of a manifest of an assembly. As is known in the art, an assembly refers to a grouping of files (modules) necessary to perform a particular application, and modules are portion(s) of a computer program that are created to carry out a particular function within the application, and can be utilized alone or combined with other modules in connection with enabling proper operation of the particular application. Shaw discloses hashing the files or modules rather than the list of files or manifest. As discussed in the specification "hashing the manifest of the referenced assembly is sufficient because that manifest in turn includes hashes of all its constituent files." (*See* pg. 3 ll. 18-20). Hashing the entire assembly would require more processing and decrease efficiency.

Evans *et al.* fails to make up for the deficiencies of Renaud *et al.* and Shaw. Evans *et al.* relates to providing versioning information for a plurality of software objects. (*See* abstract). More particularly, Evans *et al.* utilizes a hash value that is generated from the name of a version using a conventional ELF hashing function. (*See* Col. 11, ll. 46-48). The Office Action dated October 10, 2005 asserts that Evans *et al* teaches an assembly manifest referencing another assemble manifest. However, the cited art merely teaches a manifest file containing references to different modules and different versions of each module within the manifest file to account for version control. All of these references are to individual module sections within the single manifest file. Therefore, Evans *et al* does not teach or suggest an assembly manifest referencing another assemble manifest let alone hashing a manifest of a referenced assembly.

Moreover, Graunke *et al.* fails to make up for the deficiencies of Renaud *et al.*, Shaw and Evans *et al.* The Office Action attempts to reinforce the incorrect assertions that Renaud *et al.* discloses referenced assemblies and that Evans *et al.* teaches an assembly manifest referencing another assembly manifest by also citing Graunke *et al.* as teaching a hash of a referenced manifest. However, the cited art relates to a method for distributing private keys to user application programs to decrypt encrypted digital content. (*See* Abstract). The Examiner asserts that Graunke *et al.* teaches that "a list of references wherein each reference in turn contain signature information referring to other sections of a manifest". (*See* Office Action dated October 7, 2005, pg. 5). However, as described by Graunke *et al.* the manifest is a single file containing data about digital objects that may contain signatures relating to each digital object. The references refer to sections within the single manifest file, not to another assembly manifest. (*See* Fig. 3 and Col. 6 ll. 53-56). Furthermore, Graunke *et al.* teaches hashing the entire digital object, rather than the hash of a manifest or list of files of an assembly.

Therefore, contrary to assertions in the Office Action, Renaud *et al.* does not disclose referenced assemblies, Evans *et al.* does not teach an assembly manifest referencing another assembly manifest and Graunke *et al.* does not teach a hash of a referenced assembly manifest.

In view of the foregoing, the combination of the cited art fails to teach or suggest *providing an assembly manifest with a hash of a manifest of at least one other assembly that the assembly depends* as in the claimed invention. Accordingly, it is readily apparent that Renaud *et al.*, Shaw, Evans *et al.* and Graunke *et al.*, alone or in combination, do not make obvious appellants' invention as recited in independent claims 1, 10, 18, 22, 23, 27 and 30 (and

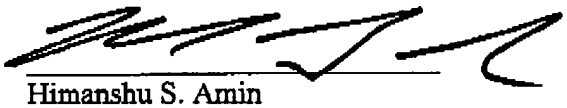
dependent claims 2-8, 11-17, 20, 21, 24-26 and 29 which respectively depend there from). Reversal of this rejection is respectfully requested.

B. Conclusion

For at least the above reasons, the claims currently under consideration are believed to be patentable over the cited references. Accordingly, it is respectfully requested that the rejections of claims 1-8, 10-18, 20-27 and 29-35 be reversed.

If any additional fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP119US].

Respectfully submitted,
AMIN & TUROCY, LLP



Himanshu S. Amin
Reg. No. 40,894

AMIN & TUROCY, LLP
24th Floor, National City Center
1900 East 9th Street
Telephone: (216) 696-8730
Facsimile: (216) 696-8731

VIII. Claims Appendix (37 C.F.R. §41.37(c)(1)(viii))

1. A method for facilitating integrity of an assembly employable by application programs during runtime, comprising:

providing an assembly with an assembly manifest that contains a list of modules that make up the assembly;

providing the assembly manifest with a hash of the contents of at least one module of the list of modules;

providing the assembly manifest with a hash of a manifest of at least one other assembly that the assembly depends on; and

comparing the hash retained in the assembly manifest with a hash of the at least one module obtained at runtime to identify whether a runtime version of the at least one module is substantially similar to a version utilized at build time of the assembly.

2. The method of claim 1, providing the assembly manifest with the hash of the contents of at least one module of the list of modules comprises providing the assembly manifest with a hash of each module of the list of modules that constitutes the assembly.

3. The method of claim 1, further comprising providing identity information in the assembly manifest.

4. The method of claim 3, the identity information comprising publisher information and version information.

5. The method of claim 1, further comprising providing a hash of the contents of the assembly at the end of the assembly.

6. The method of claim 1, further comprising determining if the contents of the assembly have been modified by determining an actual hash of the contents of the at least one module of the list of modules and comparing the actual hash with the hash of the contents of the at least one module of the list of modules residing in the assembly manifest.

7. The method of claim 6, further comprising determining if the publisher of the assembly is trustworthy if the assembly has been modified.

8. The method of claim 7, determining if the publisher of the assembly is trustworthy if the assembly has been modified comprises checking version information and publisher name information residing in the assembly manifest.

10. A method for facilitating integrity of assemblies employable by application programs during runtime, comprising:

providing an assembly with an assembly manifest that contains a list of referenced assemblies that the assembly depends on;

providing the assembly manifest with a hash of a manifest of at least one referenced assembly of the list of referenced assemblies; and

analyzing the hash provided to the assembly manifest and a second hash of the manifest of the at least one referenced assembly computed at runtime to determine whether changes have been made to the at least one referenced assembly between runtime and at build time of the assembly.

11. The method of claim 10, providing the assembly manifest with the hash of the manifest of at least one referenced assembly of the list of referenced assemblies comprises providing the assembly manifest with a hash of each referenced assembly of the list of referenced assemblies.

12. The method of claim 10, further comprising providing identity information in the assembly manifest.

13. The method of claim 12, the identity information comprising publisher information and version information.

14. The method of claim 10, further comprising providing a hash of the contents of the assembly at the end of the assembly.

15. The method of claim 10, further comprising determining if the contents of the at least one referenced assembly have been modified by determining an actual hash of the contents of the at least one referenced assembly of the list of referenced assemblies and comparing the actual hash with a hash of the contents of the at least one referenced assembly of the list of referenced assemblies residing in the assembly manifest.

16. The method of claim 15, further comprising determining if the publisher of the at least one referenced assembly is trustworthy if the at least one referenced assembly has been modified.

17. The method of claim 16, determining if the publisher of the at least one referenced assembly is trustworthy if the at least one referenced assembly has been modified comprises checking version information and publisher name information residing in the manifest of the at least one referenced assembly.

18. A computer readable medium having at least one computer executable component employable by an application program at runtime comprising:

an assembly including an assembly manifest that contains a list of at least one referenced assembly that the assembly references, a first hash of a manifest of the at least one referenced assembly, a list of modules that make up the assembly and a hash of the contents of at least one module of the list of modules, the hash of the contents of the at least one module is utilized to control which versions of the modules are employed in connection with the assembly at runtime.

20. The computer readable medium of claim 18, the assembly manifest including identity information and version information.

21. The computer readable medium of claim 18, the assembly being a dynamically linked library.

22. A computer readable medium having at least one computer executable component employable by an application program at runtime comprising:

an assembly including an assembly manifest that contains a list of at least one referenced assembly that the assembly references and a hash of the contents of a manifest of the at least one referenced assembly, the hash is compared to a second hash produced at runtime to evaluate whether the at least one referenced assembly is a same version as the at least one referenced assembly utilized at build time of the assembly.

23. A system for facilitating integrity of assemblies employable by application programs at runtime, the system comprising:

a first component that provides an assembly manifest for an assembly, the assembly manifest having a list of modules making up the assembly and a list of at least one referenced assembly that the assembly references; and

a second component that provides the assembly manifest with a hash of at least one module of the list of modules and a hash of a manifest of the at least one referenced assembly, the hash of the at least one module is compared with a hash of the at least one module generated at runtime to identify changes in the content of the at least one module.

24. The system of claim 23, further comprising a third component adapted to compare the hash of said at least one module with an actual hash value of the at least one module.

25. The system of claim 24, the assembly manifest including identity and version information and the third component adapted to determine if the assembly should be executed based on a review of the originator and version information, if the hash of the at least one module in the assembly manifest and the actual hash value of the at least one module are different.

26. The system of claim 23, further comprising a binding component adapted to provide binding policy information for determining a version of an assembly that an application program will run if another assembly having the same name resides on the system.

27. A system for facilitating integrity of assemblies employable by application programs at runtime, the system comprising:

a first component that provides an assembly manifest for an assembly, the assembly manifest having at least one referenced assembly, the at least one referenced assembly comprising a manifest;

a second component that provides the assembly manifest with a hash of the manifest of the at least one referenced assembly; and

a third component that compares the hash of the at least one referenced assembly in the assembly manifest with an actual hash value of the at least one referenced assembly to identify version changes.

29. The system of claim 27, further comprising a binding component adapted to provide the third component with binding policy information.

30. A system for facilitating integrity of an assembly employable by application programs at runtime, the system comprising:

means for relating an assembly manifest having a list of at least one related assembly to an assembly, the at least one related assembly comprising a manifest;

means for providing the assembly manifest with a hash of the manifest of the at least one related assembly; and

means for evaluating integrity of the assembly by comparing the hash value with a second hash value computed at runtime.

31. The system of claim 30, the at least one related assembly being a module.

32. The system of claim 30, the at least one related assembly being a referenced assembly.

33. The system of claim 30, further comprising means for comparing a hash value of the at least one related assembly in the assembly manifest to an actual hash value of the at least one related assembly.

34. The system of claim 30, further comprising means for establishing a binding policy.

35. The system of claim 30, at least one of the assembly and the at least one related assembly being a dynamically linked library.

IX. Evidence Appendix (37 C.F.R. §41.37(c)(1)(ix))

None.

X. Related Proceedings Appendix (37 C.F.R. §41.37(c)(1)(x))

None.